Parallel Storage Systems for Large-Scale Machines

Christos Filippidis*
cfjs@outlook.com
Department of Informatics and Telecommunications,
National and Kapodistrian University of Athens, Greece.

*Dissertation Advisor: Yiannis Cotronis, Associate Professor.

Introduction

The most important factors affecting I/O performance are [5]:

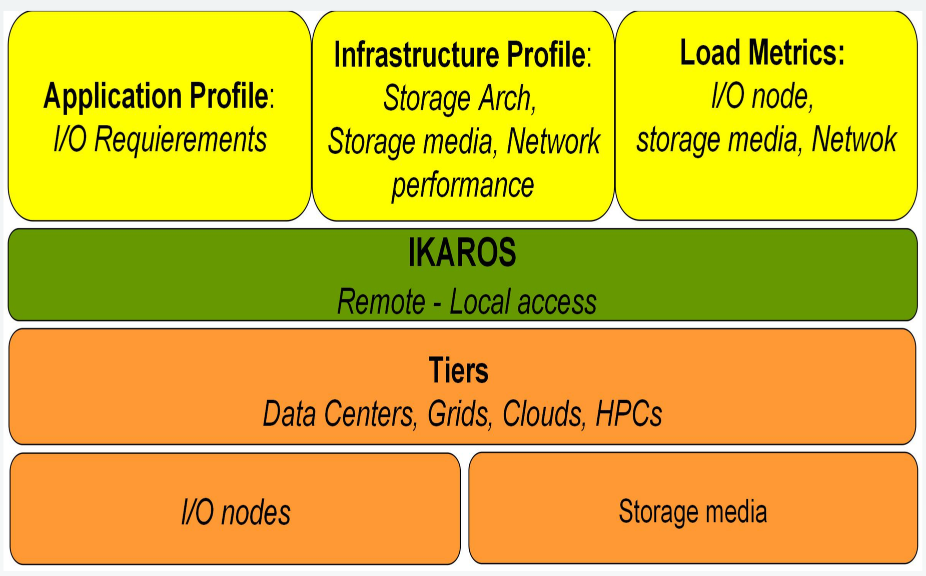
1. The number of parallel processes participating in the transfers,
2. The size of the individual transfers,
3. The I/O access patterns.

Globally shared file systems, using current storage architectures, have several performance limitations when used with large-scale systems, because:

1. Bandwidth does not scale economically to large-scale systems,
2. I/O traffic on the high speed network and on each storage server can be affected by other unrelated jobs [1].

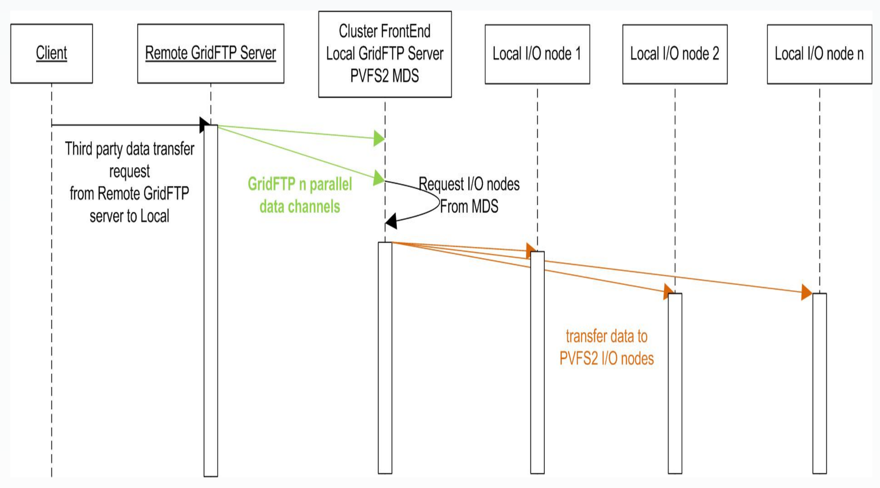
Methods

In order to confront the aforementioned problems and limitations we introduced IKAROS [3, 4]. IKAROS is a write-optimized system that provides a dynamically coordinated I/O architecture for I/O accesses according to the topology/profile of the infrastructure, the load metrics, and the I/O demands of each application.

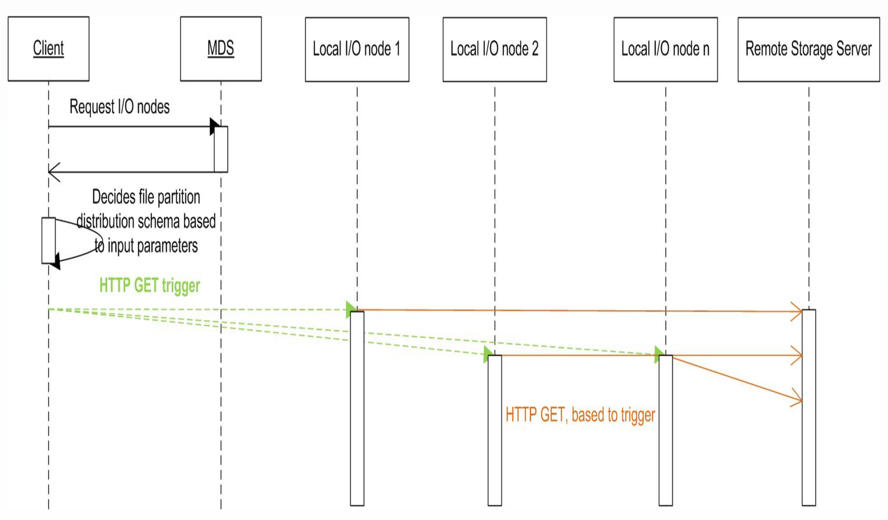


In the following figures we analyze a GridFTP+PVFS2 and an IKAROS implementation of the overall data flow.

The combined use of GridFTP and PVFS2 forces us to initiate many independent transfers incurring much overhead to set up and release connections. This approach can significantly impact performance due to the unnecessary network and disk contention.



By using IKAROS we apply only coordinated parallel data transfers in contrast with the GridFTP+PVFS2 case where we must manually synchronize the stripe size and the stripe mapping between them.

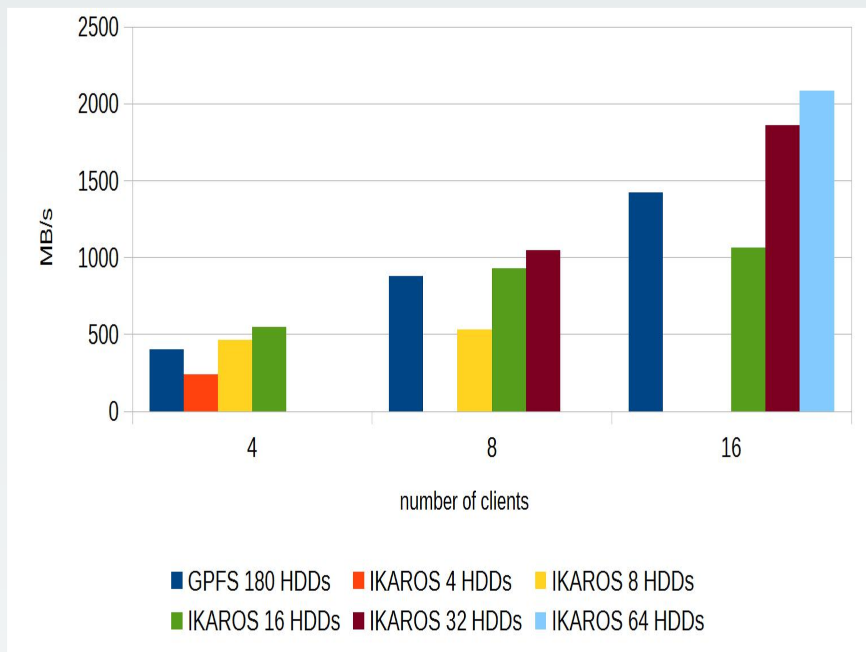


Results

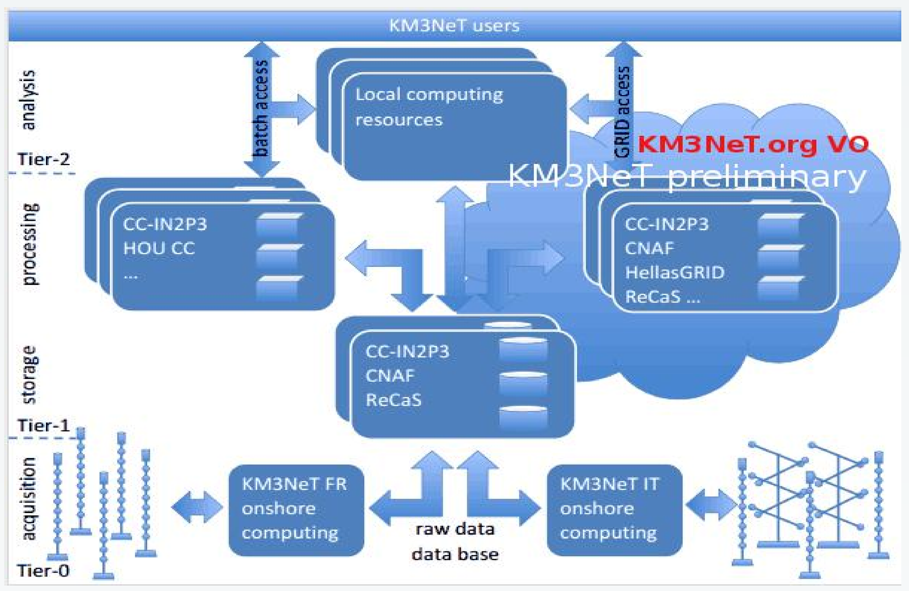
This study:

- Proposes a dynamically coordinated I/O architecture based on input parameters.
- Creates, on the fly, dedicated or semi-dedicated clusters of HDDs per job.
- Provides coordinated parallel data transfers on the overall data flow.
- Minimizes disk and network contention.
- Improves I/O performance by 33% with the 1/3 of the available hard disks.

In [2] we show how we can fully utilize the provided I/O and network resources, and improve I/O performance based on the previous mentioned remarks.

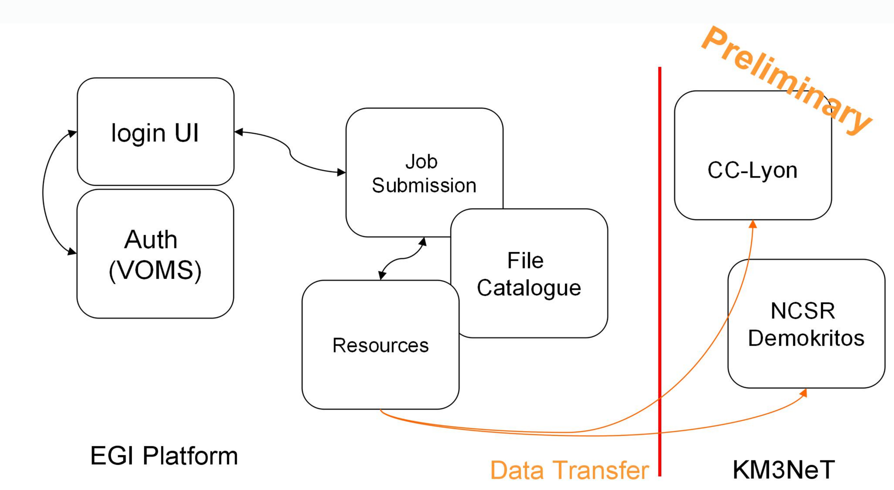


IKAROS @ KM3NeT.org



Default procedure: The data output is transferred from the Grid Worker Node to a local Storage Element (SE), from the local SE to CC-Lyon, and then at the UI, by using the GridFTP and SSH protocols (several read-write operations).

IKAROS: Sends the output directly to the destination (laptop, local computer cluster, CC-Lyon) (one read-write operation, mainly data routing) [7,8].



Conclusion

- In order to be able to scale I/O performance against both technical and economical aspects we will have to reorganize the storage architecture. Our main focus should be to achieve a balance between the available network and I/O bandwidth [2].

  New storage architecture: the actual number of HDDs per I/O node depends on the HDD I/O metrics and the network connectivity [2].

- By following the rule of max 2 writes and max 4 read requests (concurrent) per HDD, derived from [2], we are be able to fully utilize the available storage hardware while scaling the storage facility.
- By using the IKAROS reverse read technique [3], for write operations, we are able to apply only coordinated parallel data transfers for the overall data flow (remote-local access).

References

[1] Dongarra, J., Beckman, P. et al. The International Exascale Software Roadmap," , Volume 25, Number 1, 2011, International Journal of High Performance Computer Applications, ISSN 1094-3420. Exascale Nearby Storage, Cray Position paper.
[2] Christos Filippidis, Panayiotis Tsanakas,Yiannis Cotronis, IKAROS: a Scalable I/O Framework for High-Performance Computing Systems,The Journal of Systems& Software (2016),Volume 118, pp. 277-287, DOI:http://dx.doi.org/10.1016/j.jss.2016.05.027
[3] Filippidis C., Cotronis Y.,Markou C.,(2013) IKAROS: an HTTP-based distributed File System, for low consumption and low specification devices, Journal of Grid Computing springer.
[4] Filippidis C., Cotronis Y.,Markou C.,(2012) Design and Implementation of the Mobile Grid Resource Management System. Computer Science, 13 (1). pp. 17-24. ISSN 1508-2806.
[5] Schmuck, F., Haskin, R.: GPFS: a shared-disk file system for large computing clusters. In: Proceedings of the FAST 2002 Conference on File and Storage Technologies. IBM Almaden Research Center, San Jose, CA (2002).
[6] Filippidis, C., Cotronis, Y., Markou, C. (2014). Forming an ad-hoc nearby storage, based on the IKAROS and social networking services, IOP J. Phys.: Conf. Ser. 513 042018.
[7] Filippidis, C., on behalf of the KM3NeT Collaboration (1/2016), Enabling Grid Computing resources within the KM3NeT computing model, Very Large Volume Neutrino Telscope workshop (VLVnT 2015).
[8] Filippidis, C., Cotronis, Y., & Markou, C. (1/2016). Using IKAROS as a data transfer and management utility within the KM3NeT computing model, Very Large Volume Neutrino Telscope workshop (VLVnT 2015).

We acknowledge the support of Special Account for Research Grants of the National and Kapodistrian University of Athens.